An Exposé of Autonomous Agents in Command and Control Planning

Presented by:
Christopher Matthews
Claw.Matthews@us.army.mil

U.S. Army RDECOM CERDEC
Fort Monmouth, NJ
Talking Points

- Army Program
- Mission need
- Autonomous agents
  - Motivation
  - Employment
- Agent development framework
- Prototype – Maneuver Sustainment Planner
- Design & implementation of agents
- Summary
Motivation

- Logistics Command & Control (LogC2) Advanced Technology Demonstration (ATD)
  - Integrate logistics and maneuver planning
  - Faster OPTEMPO & reduced logistics footprint
  - Shorten Combat Service Support (CSS) planning times

- Achieved through the research, development and transition of:
  - Collaborative, cross-functional planning tools
  - Optimization tools for increased maneuver sustainment efficiency
  - Adaptive, predictive consumption models and demand generation functionality
  - Near real-time running-estimate decision aid software
  - Dynamic re-planning and execution-monitoring software capability
Agent Development Framework

Cognitive Agent Architecture (Cougaar)

- Darpa initiative
  - Advanced Logistics Program (ALP) (FY96 – FY01)
  - UltraLog Program (FY01 – FY04)

- Features
  - Distributed, large-scale workflow engine
  - Open source Java software
  - Multi-Resolutional Logical Data Model (LDM)
  - Built-in dynamic re-planning & execution-monitoring capability
  - Asynchronous communication protocol
  - Classic publish/subscribe blackboard
  - Domain independent
ALP Goals

Technical Goals:
- Distributed agent architecture research
- Distributed information management research
- Real-time information fusion research

Functional Goals:
- Automated logistics plan generation
- Real-time logistics situation assessment
- End-to-end movement control
- End-to-end rapid supply
Expanding the ALP vision:

- Military logistics domain
- Enhance the Cougaar framework
  - Security – trusted systems under information warfare attacks
  - Scalability – stability for large, distributed network of agents
  - Robustness – high state of survivability in chaotic environments
  - System integration – combining all of above to achieve desired systemic effects
Cougaar – Agent basics

Agent

Blackboard
(PLAN)

Publish

Subscribe

Plugins

Message Queue
Cougaar - Plugins

- **LDM**: Populating society with data from external systems.
- **Allocator**: Allocates tasks to other agents/assets.
- **Expander**: Decomposes tasks into more manageable subtasks.
- **Assessor**: Evaluates the projected and actual results of allocated tasks.
- **GUI**: Provides the web-based front end for viewing activity within society.
Agent Roles

<table>
<thead>
<tr>
<th>Agent</th>
<th>Role</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bn_1-8_Infantry</td>
<td>MobileRefuelCustomer</td>
<td>CombatUnit</td>
</tr>
<tr>
<td>Co_A_1-8_Infantry</td>
<td>MobileRefuelCustomer</td>
<td>CombatUnit</td>
</tr>
<tr>
<td>Co_B_1-8_Infantry</td>
<td>MobileRefuelCustomer</td>
<td>CombatUnit</td>
</tr>
<tr>
<td>Co_C_1-8_Infantry</td>
<td>MobileRefuelCustomer</td>
<td>CombatUnit</td>
</tr>
<tr>
<td>Bn_1-68_Armor</td>
<td>MobileRefuelCustomer</td>
<td>CombatUnit</td>
</tr>
<tr>
<td>Co_A_1-68_Armor</td>
<td>MobileRefuelCustomer</td>
<td>CombatUnit</td>
</tr>
<tr>
<td>Co_B_1-68_Armor</td>
<td>MobileRefuelCustomer</td>
<td>CombatUnit</td>
</tr>
<tr>
<td>Co_C_1-68_Armor</td>
<td>MobileRefuelCustomer</td>
<td>CombatUnit</td>
</tr>
<tr>
<td>Base_Support_Company</td>
<td>MobileRefuelProvider</td>
<td>LogisticsUnit</td>
</tr>
<tr>
<td>Forward_Support_Company</td>
<td>MobileRefuelProvider</td>
<td>LogisticsUnit</td>
</tr>
</tbody>
</table>

Agent Relationships

- Superior
- Subordinate
- Provider
- Customer
Cougaar – Re-planning Concept

Dynamic Re-planning & Execution Monitoring

Self-Assessment
Real-Time Data Fusion

Data Source

Task Allocations
Allocation Results
Rescinds / Reallocations
Requirements
- Support over 6 million items
- Support efficient transport and distribution of objects
- Support modification and extension during execution

Principles
- Based on the properties of objects and not what they are
- Represent all the properties of assets needed to reason about them
  - Over a range of granularities
  - Supporting their time-varying nature
  - Refering to specialized properties of assets

Implementation
- Use prototypes and delegation to reduce classes needed
- Prototype classes determine the required properties of all instances
- Related properties are collected in Property Groups
- Asset instances delegate properties to their Prototype instances
Asset Prototypes and Property Groups

LDM Classes

Asset Instances

NIIN= 123456789
VehicleProp= ContainerProp= OtherProp

maxSpeedMPH= fuelUseMPG= fuelType=

NIIN= 123456999
VehicleProp= ContainerProp

5-Ton Truck
Prototype

maxWtSTON= MaxVolMTON=

NIIN= 123456999
VehicleProp= ContainerProp

2.5-Ton Truck
Prototype

maxWtSTON= MaxVolMTON=

NIIN= T789
VehicleProp= ContainerProp

5-Ton Truck
Prototype

maxWtSTON= MaxVolMTON=

NIIN= T123
VehicleProp= ContainerProp

2.5-Ton Truck
Prototype

maxWtSTON= MaxVolMTON=

Asset - 5-Ton Truck With Long Bed

Truck-T770

NIIN= "T770"
ContainerProp

MaxWtSTON= 8.0
MaxVolMTON= 6.0

NIIN= "T789"

MaxWtSTON= 6.0
MaxVolMTON= 4.0

NIIN= "T123"

MaxWtSTON= 3.0
MaxVolMTON= 2.0

NIIN= T770

VehicleProp= ContainerProp

NIIN= "T789"

VehicleProp= ContainerProp

NIIN= "T123"

VehicleProp= ContainerProp
Develop proof-of-concept prototype
- Integrate logistic-planning impacts into the maneuver planning process
- Develop a *detailed* logistics plan to support maneuver operations
- Model maneuver activities and generates expected logistics demand as a function of platform, posture, and optempo
- Provides dynamic re-planning & execution-monitoring capability

Evaluate Cougaar and agent benefits for C2 planning
MSP – Why Cougaar?

- Core planning capabilities included
  - Dynamic replanning/execution monitoring
  - Resource management (asset scheduling)
- Rapid software development
- Leverage existing logistics software components
- Agents map elegantly to military force structures
- Digitization of reusable, intricate, and highly complex business models
- Easy to introduce external data into agent society
MSP - Approach

- Functional analysis & design
  - Agent Enumeration
  - Role/Relationship Analysis
  - Plugin Enumeration
  - Task Grammar
  - Asset/Property Requirements Analysis
  - Execution Monitoring and Dynamic Replanning Analysis

- External system interfaces
  - Databases
  - Maneuver Command & Control (MC2) application
  - MSP Plan Viewer (GUI)

- Cougaar Plugin development
MSP – Preliminary System Overview

Maneuver Command & Control Application (MC2)

Digitized Maneuver Plan

Feedback Loop (Feasibility, Alerts, & Recommendations)

Maneuver Sustainment Planner (MSP)

Detailed Logistics Plan
MSP – Constraints

- No control over the maneuver plan (read only)
  - Maneuver Command & Control (MC2) system
    - Stove-piped system
    - Rigid, closed plan representation
    - Large, unwieldy XML plan data

- Atypical Cougaar use
  - Driven by MC2
  - Short-lived vs. 24x7

- Demand generation (simulate consumption)
  - CASCOM Equipment Usage Profiles (EUP)
    - Equipment type
    - Optempo
    - Same as MC2
MSP – System Overview Final

Maneuver Sustainment Tool (MSP)

Digitized Maneuver Plan

Databased Plan

Detailed Logistics Plan

Feedback Loop
(Feasibility, Alerts, & Recommendations)
In Summary

Pros
- Rapid software development
  - Provides logical roadmap for application design & development
- Ideal for military planning systems
- Domain independent
- Open source software
- Core planning & information management functionality
- Well documented architecture & developer guides

Cons
- Large overhead
- Bandwidth intensive
- Steep learning curve
- Poorly documented software (sparse Javadocs)
- Frequent Architecture upgrades
Backup Slides
Cougaar + UltraLog Society

**Cougaar**
- Basic building blocks
- Easy to specialize
- Domain independent

**Generic Agent**

**Generic Plugins**

**UltraLog Society**

- Military specific processes
- Interfaces to military systems
- Specific to Logistics Domain

**Domain Agent**

- Allocator
- Expander
- Assessor
Cougaar – The Distributed Plan

Agent

Blackboard

Asset

Task

Plan Element (Allocation)

Estimated AR

Reported AR
Plan Element - Allocation

Task-1

Plan Element (Allocation)

Asset

Estimated AR

Reported AR
Plan Element - Expansion

Task-1

Plan Element (Expansion)

Workflow

Subtask-1.1

Subtask-1.2

Estimated AR

Reported AR
All Assets in the logc2.assets package are machine generated from the logc2props.def and the logc2assets.def files. All ground vehicle Assets will be of one of the types depicted here.

This class is hand generated (the only one in the package.) This is needed so that we can add functionality later to all Assets w/o having to mod lots of classes.
References

- Barger, Mark, & Wong, Jason. (2004). *Cougaar Training Slides*.